

Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A method comprising:

determining by a first execution thread whether if a user is proximately located with respect to a device;

setting an indicator, by the first execution thread, if it is determined that the user is proximately located with respect to the device;

determining by a second execution thread, different from the first execution thread, whether if there is activity on the device;

re-setting the indicator, by the second execution thread, if it is determined there is activity on the device within a first time period; and

as long as the user is proximately located, and there is no activity on the device, periodically simulating by the second execution thread, an activity on the device to prevent a first resource of the device from transitioning into a first resource saving state, if it is determined that there is no activity on the device within the first time period, and the indicator remains set.
2. (Currently Amended) The method of claim 1 wherein said determining by the first execution thread whether if the user is proximately located comprises monitoring by the first execution thread an audio input device for audio input.

3. (Currently Amended) The method of claim 2, where said determining ~~if~~by the first execution thread whether the user is proximately located further comprises determining ~~if~~whether the user's voice is present in said audio input.

4. (Currently Amended) The method of claim 3, where said determining ~~if~~by the first execution thread whether the user is proximately located further comprises comparing audio samples from said audio input to a voice reference sample of the user.

5. (Currently Amended) The method of claim 1, where said determining ~~if~~by the first execution thread whether there is activity on the device comprises receiving notification of activity from an operating system of the device.

6. (Currently Amended) The method of claim 5, where said determining ~~if~~by the first execution thread whether there is activity on the device further comprises requesting said operating system to provide said notification of activity.

7. (Currently Amended) The method of claim 1 wherein said ~~period for simulating said activity~~first time period has a period length shorter than a period of inactivity that will result in the first resource of the device in entering said first resource saving state.

8. (Original) The method of claim 1 wherein said simulating of activity comprises simulating one or more of a key press, a pointer device movement, and a network traffic event.

9. (Currently Amended) An apparatus comprising:

storage medium having stored therein a plurality of programming instructions designed to implement a first execution thread equipped to determine if ~~whether~~ a user is proximately located with respect to the apparatus, and set an indicator if it is determined that the user is proximately located with respect to the device; and a second execution thread equipped to determine if ~~whether~~ there is activity on the apparatus, re-set the indicator if it is determined there is activity on the apparatus within a first time period, and simulate an activity to prevent a first resource of the device-apparatus from transitioning into a first resource saving state if it is determined that the user is proximately located and there is no activity on the apparatus within the first time period, and the indicator remains set; and

a processor coupled to the storage medium to execute the programming instructions.

10. (Currently Amended) The apparatus of claim 9, wherein said programming instructions are designed to equip the first execution thread to perform said determining ~~whether~~ if the user is proximately located by monitoring an audio input device of the apparatus for audio input.

11. (Currently Amended) The apparatus of claim 10, where said programming instructions are designed to equip the first execution thread to determine if the user's voice is present in said audio input, when performing said determining if ~~whether~~ the user is proximately located.

12. (Currently Amended) The apparatus of claim 11, where said programming instructions are designed to equip the first execution thread to compare audio samples from said audio input to a voice reference sample of the user, when performing said determining if-by the first execution thread whether the user is proximately located.

13. (Currently Amended) The apparatus of claim 9, where said programming instructions are designed to equip the first execution thread to receive notification of activity from an operating system of the apparatus, when performing said determining if-whether there is activity on the apparatus.

14. (Currently Amended) The apparatus of claim 13, where said programming instructions are further designed to equip the second execution thread to request said operating system to provide said notification of activity, when performing said determining if-whether there is activity on the apparatus.

15. (Currently Amended) The apparatus of claim 9, wherein said first time period~~period for simulating said activity~~ has a period length shorter than a period of inactivity that will result in the first resource of the apparatus in entering said first resource saving state.

16. (Currently Amended) The apparatus of claim 9 wherein said programming instructions are designed to equip the second execution thread to simulate one or more of a key press, a pointer device movement, and a network traffic event.

17-24 (Canceled).

25. (New) The method of claim 1, wherein the second execution thread is spawned by the first execution thread.

26. (New) The method of claim 1, wherein said determining by the second execution thread whether there is activity on the device comprises determining by the second execution thread at least one of determining whether there is a key press, determining whether there is a pointer device movement, and determining whether there is a network traffic event.

27. (New) The method of claim 1, further comprising:

determining by a third execution thread, different from the first and second execution threads, whether there is activity on the device;

re-setting by the third execution thread, the indicator if it is determined there is activity on the device within a second time period; and

simulating by the third execution thread, an activity on the device to prevent a second resource of the device from transitioning into a second resource saving state, if it is determined that there is no activity on the device within the second time period, and the indicator remains set, the first and second resources being different resources.

28. (New) The method of claim 27, wherein the second and third execution threads are spawned by the first execution thread.

29. (New) The apparatus of claim 9, wherein said programming instructions are designed to equip the first execution thread with an ability to spawn the second execution thread.

30. (New) The apparatus of claim 9, wherein said programming instructions are designed to equip the second execution thread to perform said determining of whether there is activity on the device by determining at least one of whether there is a key press, whether there is a pointer device movement, and whether there is a network traffic event.

31. (New) The apparatus of claim 9, wherein said programming instructions are further designed to implement a third execution thread equipped to determine whether there is activity on the device, re-set the indicator if it is determined there is activity on the device within a second time period, and simulate by the third execution thread, an activity on the device to prevent a second resource of the device from transitioning into a second resource saving state, if it is determined that there is no activity on the device within the second time period, and the indicator remains set .

32. (New) The apparatus of claim 31, wherein said programming instructions are further designed to equip the first execution thread with an ability to spawn the second and third execution threads.